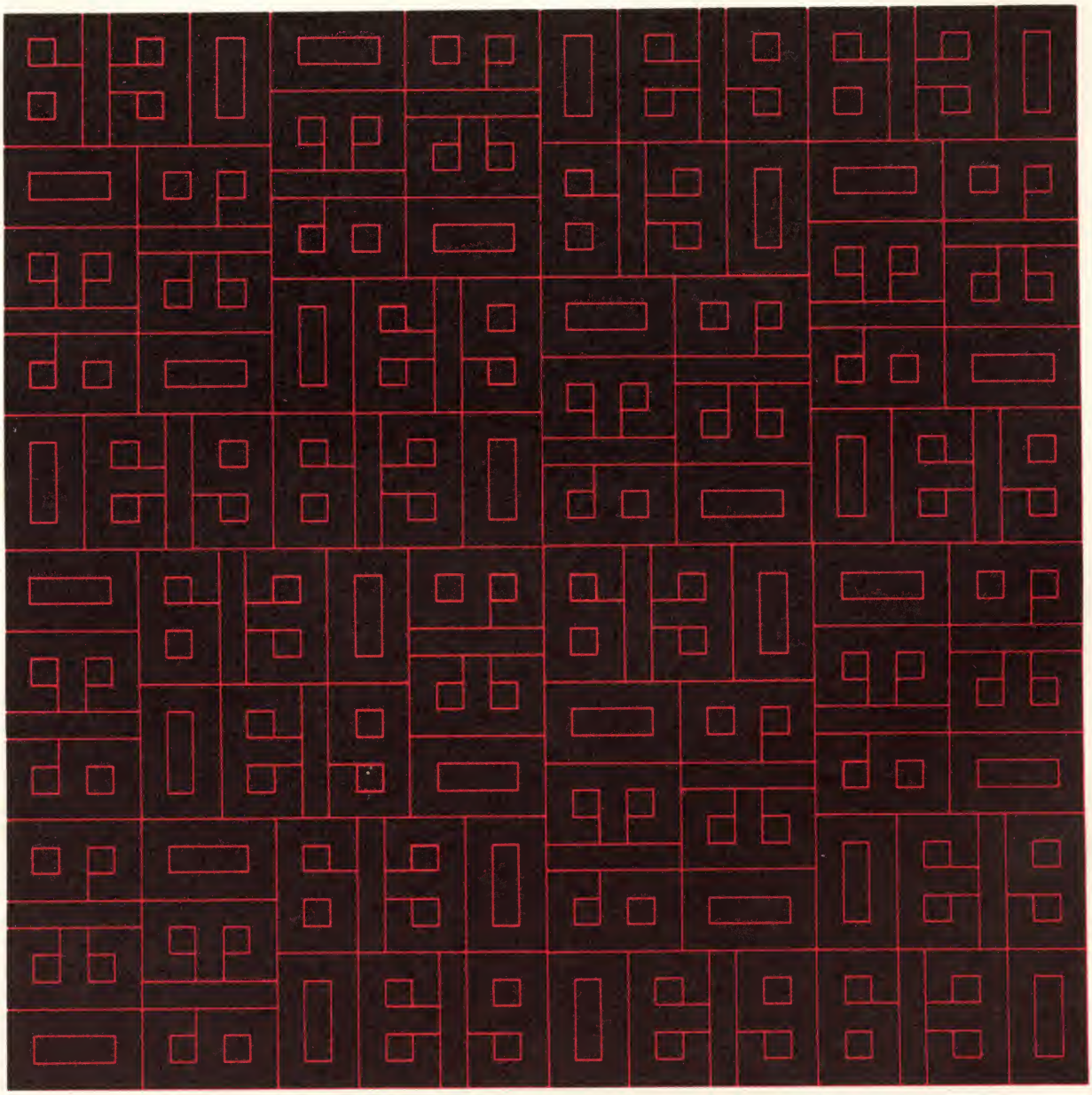
ADVANCE 6130 COMPUTING SYSTEM

Preliminary Description

Form No. P-1A 6-30-66



EMR ASI COMPUTER DIVISION

ADVANCE 6130 CENTRAL PROCESSING SYSTEM

PRELIMINARY DESCRIPTION

FORM NO. P-1A

TABLE OF CONTENTS

ADVANCE 6130 PRELIMINARY DESCRIPTION

SECTION ONE GENERAL DESCRIPTION

The ADVANCE 6130 is a high-speed, integrated circuit, parallel-binary data processor. It has flexible data acquisition and system control features provided by a large set of fast, powerful instructions for computation and data manipulation. Communication with peripheral devices is by programmed input/output, buffered sequential word, or multiplexed buffered channels. The ADVANCE 6130 has memory options that vary in speed, size, and multiple access capability.

An ADVANCE 6130 system is composed of the following:

- One or more processor modules, each with a Programmed Input/Output Channel
- One or more memory modules with one or more interface buses
- One or more operator control panels (attached or remote) for each processor
- Zero to six buffered channels (sequential Word/Byte or Word Multiplex) for each memory bus
- One optional control teletype for each processor
- Flexible sense, signal, and multilevel priority interrupt options
- Various standard or special peripherals
- Optional real-time clock

Each system includes the combination of modules that fit its application. All modules are made to mount in standard ASI racks. A system requiring only a few options will fit into a single cabinet and a system with many options will require two cabinets.

1.1 Processor

The processor does the computation, instructs peripheral devices, controls the Programmed I/O Channel directly, sends external signals, tests external sense lines, and responds to external interrupts. All program operations are implemented by the processor.

The processor communicates with all memory modules that are connected with the common memory bus. The control panel connects to the processor which implements all of its controls and displays. The processor instructs peripheral devices by using the programmed I/O channel. This channel is also used for direct program controlled input and output of data, sending external signals, and testing external sense lines. Up to 16 optional levels of external priority interrupt, each of which may be separately allowed or disallowed, are provided in the processor module. Additional priority interrupts, up to a total of 128, are available when using optional priority interrupt modules. Optional sense and signal modules use the programmed channel interface to select groups of 8 or 16 unitary coded sense or signal lines with each I/O instruction. Special sense and signal systems may use the programmed I/O interface directly.

The arithmetic unit is physically located in the processor. This includes registers **A** and E which are 16-bit programmable arithmetic registers. Numbers are in the two's complement system. Bit 16 is the sign and Bit 15 is the most significant bit.

Program control is also located in the processor. This includes three hardware index registers and the P register which specifies the address of the next instruction. Single indexing does not delay instructions that come from a memory with a 0.9 microsecond cycle time or longer. Double indexing will delay an instruction less than 200 nanoseconds. Multi-level indirect addressing delays an instruction by one memory cycle per level. Instructions may be in the short (1 word) or long (2 words) format.

Some typical instruction times for the ADVANCE 6130 processor are shown below:

	0.9 Microsecond Memory
Add (short format, indexed)	1.8*
Load A (short format, indexed)	1.8
Store A (short format, indexed)	1.8
Multiply (short format, indexed)	3.6 to 5.9

Jump (short format, indexed)	0.9
Add (long format)	2.7
Multiply (long format)	4.5 to 6.8
Shift Operations	1 + n(0.14); n = Shift count
Divide	8.1
Logical operation	1.8
Jump and clear interrupt	0.9
Jump if a greater than zero	2
Skip a low	2
Increment and test	3
Store return address and jump	2
Set memory protect	2

* All times are given in microseconds.

All of these instructions, except shift, will operate faster when a faster memory is installed.

Each priority interrupt is assigned a fixed address in lower memory which contains the address of the start of the corresponding subroutine. When an interrupt occurs the instruction at the start of the subroutine (its location is contained in the interrupt fixed location) is executed. If this instruction is an SST, RTJ, or JMP the appropriate interrupt level will be set and the subroutine will be entered. Otherwise control will be returned to the original program at the point of interruption unless the executed instruction specifies otherwise. Each level of priority interrupt can interrupt lower levels. The following interrupts are provided:

Interrupt	Priority	Interrupt Address
Power fail	1	00002
Parity fail (on internal data transfer)	2	00003
Add overflow or underflow	3	00004
Memory protect violation	4	00005
External priority interrupts	as spec	00100-00275
Programmed I/O	2nd lowest	00006
Scanned interrupt	lowest	00007

Other fixed memory address assignments are:

Programmed instructions	00040-00077
Buffered channels 1 - 6	00010-00037
Multiplexer channel (if there is one)	00300-00377
Monitor Mode entry	00000-00001
Protected Push	00276
Unprotected Push	00277

The ADVANCE 6130 instruction set is listed in Section II of this document. The following are some of the features of this instruction set:

- 32 programmed instructions
- Three index registers, single or double indexing
- Short or long instruction format
- Multi-level indirect addressing
- Hardware multiply and divide
- Many types of shift including normalize, tally, and transpose
- Many test instructions including bit test
- Programmed input to A or memory
- Programmed output from A or memory

The ADVANCE 6130 has 15-bit address registers. This permits direct random addressing of up to 32,768 words of memory.

1.2 Memory Modules and Memory Interface

Each memory module contains the memory elements, memory read/write circuitry, and interface logic for a complete memory system. Memory modules may vary in type, size, and speed but they all use the same standard memory interface. The asynchronous nature of this interface permits several relatively asynchronous users (processor, buffered channels, or external devices) to efficiently share memories of various speeds. Each memory responds as fast as its speed permits.

Each memory module may connect with 0 to 7 users on each of two independent memory buses. Each user is assigned a priority relative to other users on the same memory bus. Memory service is on a first-come, first-serve basis. The priority structure decides cases of simultaneous memory

access requests. The second memory bus permits users on different memory buses to obtain simultaneous access to different memory modules.

Each memory word length is 18 bits. Included is a memory protect bit, a parity bit, and 16 data bits. If the memory protect bit is "0", the word may be freely accessed for read or write operations. If the memory protect bit is "1", read references are allowed but write references are prevented except by protected instructions or buffered channels. In the event of an illegal attempt to write in a protected location, the original contents of that location will be restored and a memory protect violation interrupt will occur. Memory protect bits may be set or cleared by the DLD instruction when the program is in the monitor mode (MM flip-flop is set). The MM flip-flop may be set only with an accompanying unconditional transfer of control to location 00000 in the monitor program. If this location is protected, no program except the monitor can change memory protect status. Memory protect status can be changed from the operator control panel when the ADVANCE 6130 is halted in the ONE-INSTRUCTION or BREAK-POINT mode. Memory protect status can be changed by certain privileged external devices such as the mass memory.

Parity, which is stored in each memory location, is transferred with all intermodule communication. Parity checking for common bus memory references is accomplished in the processor.

Memory modules of 4096 or 8192 words, both having a speed of 0.9 microsecond, are offered for the ADVANCE 6130. Other sizes may be offered in the future. A total of up to 32,768 words of memory may be included in an ADVANCE 6130 system. The entire memory is directly addressable.

1.3 Operator Control Panel

The Operator Control Panel is used for control of the ADVANCE 6130. It contains the following display lights, entry switches, and control switches.

POWER ON/OFF - turns power on or off. Light indicates on.

EMERGENCY CLEAR - optional - clears A, E, P, X1, X2, X3, and various control flip-flops. Emergency clear also arms preset.

4 SENSE SWITCHES - may be tested by the I/O instruction using ED address 00 (octal). The I/O instruction gets a busy response on a control operation to address 00 if a specified sense switch is in the "One" state. If more than one bit is a "One", the test is an "Or" function of the specified sense switches.

SENSE SWITCH 1 CORRESPONDS TO BIT 1
SENSE SWITCH 2 CORRESPONDS TO BIT 2
SENSE SWITCH 3 CORRESPONDS TO BIT 3
SENSE SWITCH 4 CORRESPONDS TO BIT 4

PRESET - optional. When the preset option is in, this switch causes the first few locations in memory to be loaded with a bootstrap program. Preset is armed by emergency clear.

2 STATUS LIGHTS - controlled by the I/O instruction using ED address 00 (octal). When a control operation is addressed to 00, bit 5 of the data word specifies LIGHT 1 and bit 6 specifies LIGHT 2. Bit 7 of the data word specifies setting or clearing or either or both lights.

RUN MODE SWITCH - specifies the operating mode of the ADVANCE 6130. The following three mode settings are possible:

CONTINUOUS - used for ordinary operation.

ONE INSTRUCTION - causes the program to halt at the end of the current instruction.

BREAK POINT - causes the program to halt before executing any protected instruction.

18 DATA INDICATORS - display the contents of various specified registers. Any memory location can be displayed. The parity and memory protect bits are included but are only meaningful in the case of memory display.

17 DATA SWITCHES - may be used to specify a word of data plus memory protect which may be loaded into various registers or any memory location. The memory protect bit applies only to memory references.

DISPLAY - ENTER MODE SWITCH - specifies the mode of display or entry. The following three settings are possible:

SCAN - Increments memory address after each display or enter memory operation.

REGULAR - Specified display or entry occurs once with memory address unchanged in memory display or entry.

CYCLIC - Specified display or entry occurs repeatedly while switch is set.

8 DISPLAY SWITCHES - armed only when the processor is halted. These switches cause display of A, E, P, X1, X2, X3, MA, or memory.

8 ENTER SWITCHES - armed only when the processor is halted. These switches cause the contents of the data switches to be entered in A, E, P, X1, X2, X3, MA, or memory.

CONSOLE DISARM FEATURE - In order to facilitate the operation of the ADVANCE 6130 in a real time environment, the console contains a key switch whose function is to disarm the console display and data entry operation. In the disarmed state all switches and indicators are inoperative. When the console is in a disarmed state, it may be physically removed from the processor unit.

1.4 Buffered Input/Output

The ADVANCE 6130 contains as many as six buffered data channels, one of which may be a Word Multiplex Channel with as many as 16 subchannels. Buffered channel and channel adapters presently available include:

1. Word Multiplex Channel - word oriented, bidirectional 1 to 16 subchannels, handles words or 6 or 8 bit bytes.
2. High Speed Word/Byte Channel - bidirectional, memory rate communications.
3. Six Bit Character-Assembling Channel - compatible with ADVANCE 6000 Series peripherals.

The external interface of the Word Multiplex and Word/Byte Channels are compatible with each other. This permits external devices and adapters to be connected to any of these types of channels.

Buffered channels connect internally to a memory bus and externally to the external devices which use them. The ADVANCE 6000 Series channel receives its control instruction and interrupt scan from the Programmed I/O Channel cable.

External device operations are initiated by the I/O instruction from the processor module. The external device control logic has the timing initiative for communication with data channels.

All channels can communicate in variable-sized blocks. The memory address limits for each block are specified by the current address (BM) and the limit address (BL). BM is incremented by one during each memory reference until it becomes equal to BL.

Parity is transmitted with all data transfers and may be checked in the external device on output. It may be generated by the external device or by the channel on word input. However, character parity is always generated by the external device and checked by the channel which also supplies the parity of the assembled word.

The Word/Byte Channel or the Word Multiplex subchannel are capable of single block communication or continuous single/multiple block cyclic communication. Four fixed memory locations are assigned to each word/Byte Channel and Word Multiplex subchannel. These locations contain BMO, BLO, BM1, and BL1 for that channel. BM and BL are the current and limit addresses, respectively, for the memory data blocks which are assigned to the channel by the program. For single block operations the program need only load the BMO and BLO addresses. For cyclic operation the channel alternately uses BMO and BLO then BM1 and BL1 for its data reference addresses. Whenever a block is completed in cyclic operation the external device may interrupt the processor to cause the BM and BL addresses to be updated by the program for the block following the next memory block. The external device determines when the end of a block has been reached by looking at the stop line (BM = BL) or by some characteristic of the data.

1.4.1 Word Multiplex Channel

The Word Multiplex Channel communicates with from 1 to 16 external devices in a word-multiplex mode. Each external device or adapter that communicates with the Word Multiplex Channel has its own subchannel number. In the normal multiplex mode the channel makes the following memory references for each word of data.

1. Obtain BL (limit address) value.
2. Obtain BM (current address) value.
3. Transfer data to or from storage.
4. Restore previous BM value (after incrementing).

Steps 1 and 2 occur only if the current request is from a different subchannel than the previous request.

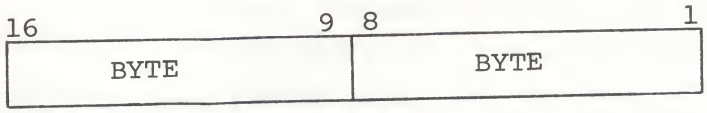
The word Multiplex Channel will operate in the two cycle mode when consecutive channel references are from the same subchannel. In this mode, only the data transfer and memory address updating memory references occur for each word of data. All subchannels are bidirectional. Each subchannel has its own level of priority which is equal to its subchannel number.

The Word Multiplex Channel is mainly intended for slow or medium speed communication. Under restricted conditions, it can be used for high speed communication.

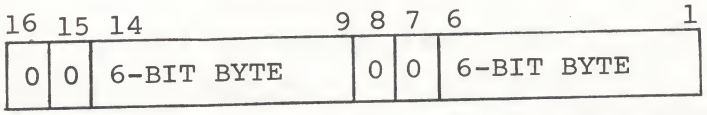
1.4.2 Word/Byte Channel

The Word/Byte Channel communicates with from 1 to 16 external devices in a block multiplex mode. It is mainly intended for high speed data communications. This channel is capable of communicating at up to memory cycle rates. The Word/Byte Channel uses only one memory reference for each word of data transferred. When the channel is initialized, it uses two additional memory references to obtain the BM and BL Values. The Word/Byte Channel can be used for cyclic or single block communication and is bidirectional. Only one device at a time may use the word channel.

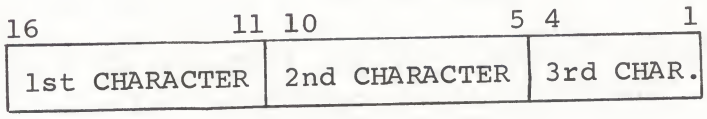
The Word/Byte Channel assembles 6 or 8 bit bytes into words on input. It disassembles words into 6 or 8 bit bytes on output. The byte formats are as follows:



8-Bit Byte Format (Alphanumeric or Binary)



6-Bit Alphanumeric or Unpacked Binary Format



6-Bit Packed Binary Format

The program specifies formats to external device controllers by means of the I/O instruction. The device controller in turn specifies the format to the Word/Byte Channel.

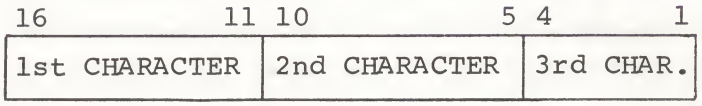
External devices on the Word/Byte Channel use the channel busy line to notify other devices that the channel is already in use.

1.4.3 ADVANCE 6000 Series 6-Bit Character Channel

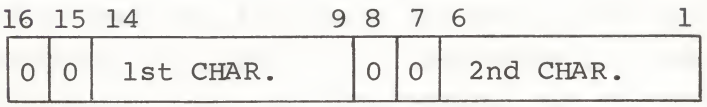
The ADVANCE 6000 Series Character Channel connects to a memory bus, programmed channel, and with up to 16 external devices. It communicates with the memory in 16-bit words and with peripheral devices in 6-bit characters.

Characters are assembled or disassembled in one of the following 2 formats:

1. Packed Binary



2. Alphanumeric



This channel can accommodate ADVANCE 6000 Series peripheral equipment.

The program instructs the channel (via the I/O instruction) which format to use when it initializes the channel. The channel also transmits instructions to the external devices and scanned interrupts to the 6130 by the programmed channel.

1.5 Memory Protect Operation

The ADVANCE 6130 computer word is 18 bits in length. Bit 18 of each memory word is used to specify the protected status of that word. A 1 in the Bit 18 position implies that the word is protected.

The use of the protect bit is to prevent writing into a particular memory location by an unprotected instruction. Any instruction can read any memory location regardless of the protect status of the object memory location. Any protected instruction can write in any memory location regardless of the protect status of the object memory location. Unprotected instructions (i.e. those instructions stored in locations whose memory protect bit is not set) can write in any other unprotected memory locations. In the long format (2 words) a protected instruction is defined by the presence of a one in the memory protect bit of its first word. A memory protect violation interrupt will result whenever an unprotected instruction attempts to write in a protected memory location.

In order to insure the orderly flow of program control between protected and unprotected instructions, a memory protect violation interrupt will result whenever the execution of an unprotected instruction is followed by the execution of a protected instruction without an intervening interrupt. The same result will occur if an unprotected XEC calls a protected instruction. In this case the protected instruction will not be executed and the memory protection violation interrupt will occur.

1.5.1 Memory Protect Alterations

Alteration of the memory protect status of any memory location is under control of the Monitor Mode (MM) Flip-Flop. Any program attempt to alter the protect status of the memory location when the Monitor Mode (MM) Flip-Flop is not in the enable condition will result in a memory protect violation interrupt with the contents of the subject memory location including its status remaining unchanged. The Monitor Mode (MM) Flip-Flop may be enabled (set to a 1 state by the execution of a SMM) set Monitor Mode (MM) command. The execution of the SMM command results in a transfer of control to location 00001, regardless of the operand address specified by the SMM command.

1.5.2 Protected I/O Operations

The ADVANCE 6130 memory protection permits the designation of specific external devices to be designated as protected. This is implemented in the following manner: the protect bit for the Programmed I/O instructions which is used to initiate data transfers is transmitted on the Programmed I/O cable during the execution of the I/O command. Standard ADVANCE 6130 external devices are equipped with a two position switch which controls the responses to the I/O command. If the switch is in position A, the device responds to any I/O command which addresses it regardless of the protect bit of the I/O command. If the switch is in position B, the device will respond "busy" to any I/O instruction that addresses the device, unless the I/O command protect bit ia a 1. Thus, standard devices can be set to ignore unprotected instructions. Special devices can be made to always ignore unprotected instructions. Devices which are to be

used only by background programs can be set to accept unprotected instructions. Some external devices (such as mass memory) may set or clear memory status along with data storage operations.

All ADVANCE buffered data channels are capable of transmitting the memory protect status of a data word or byte, to or from the memory. In general, the buffered data channels will read or write into memory locations, ignoring the protection status of the memory location involved. Two control lines contained in the buffered data channel cable determine the mode of operation of the channel with respect to the memory protection status.

The first control line is called the set protect line. If this line is set to a 1 during the data transmission the protect bit of the word being stored will be set. The second control line is the clear protect line. If this line is set to a 1, the memory protect bit of the word being transmitted will be cleared. If both lines are set to a zero, the memory protect status of the location to which data is being transmitted will be ignored. It should be noted that while the protect bit is ignored for the purpose of data transmission it is not altered, unless the device is specifically instructed to alter it, and if it was previously set to 1, the stored data word will be in a protected state.

1.6 Status Operations

In order to allow the easy implementation of multiple real time user programming, the ADVANCE 6130 contains two commands which load and store the status of the computer. These commands load or store the program counter, accummulator, extension register and the three index registers in a push-down stack in core memory. The present level of the push-down stack is stored in a unique memory location which is referred by each store status or load status command. This unique memory location is called the "Push" location. There are two unique Push locations in each ADVANCE 6130. One location responds only to protected store status and load status instructions. The other is used for all unprotected store status and load status instructions. A memory protection violation interrupt will result if the address contained in the protected Push location is in an unprotected state. Alternatively, a memory protect violation inter-

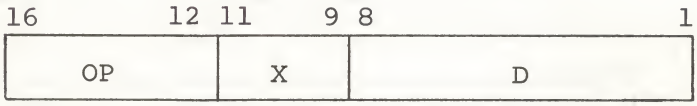
rupt will occur if the address contained in the unprotected location is in a protected state. This permits the program to detect when the limits of the push-down store have been exceeded. The same push-down store can also be used by reentrant subroutines for dynamic storage.

1.7 Optional Teletype

A standard Model 33 or Model 35 teletypewriter can be attached to an ADVANCE 6130. Interface logic for this teletypewriter is included in the processor module. The teletypewriter communicates with the processor through the Programmed Channel.

SECTION TWO INSTRUCTION SET

2.1 Short Format



Bits 16 - 12 contain the instruction code specification. This allows 31 short format opcodes with the 32nd opcode (11111) being reserved for specifying extended format which is discussed in Paragraph 2.2.

Bits 11 - 9 are used for specification of address modification. The capabilities are specified as follows:

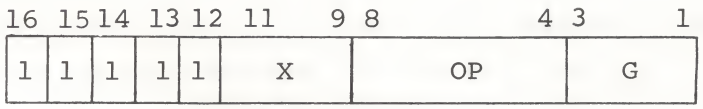
X	Meaning
0	EA = P + D
1	EA = X1 + D
2	EA = X2 + D
3	EA = X1 + X2 + D
4	EA is obtained via indirect through P + D
5	EA is obtained via indirect through X1 + D
6	EA = X3 + D
7	Immediate Addressing $(0 \leq D \leq +255_{10})$

Note: EA = Effective Address
P = Program Sequence Register
Xn = Index Register n
$-128_{10} \leq D < +128_{10}$ except for immediate addressing

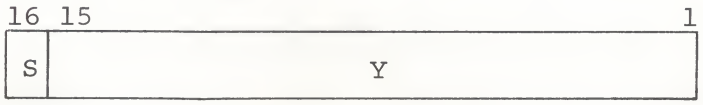
In all cases other than designator specification or immediate addressing, negative D is considered a two's complement binary number. When immediate addressing is specified, D represents 8 bits of magnitude with the most significant bit positions (16 - 9) being zero.

2.2 Long Format

Word 1



Word 2



Bits 16 - 12 contain ones to specify long format. Bits 11 - 9 are analogous to short format specifications but are interpreted as follows:

X	Meaning
0	EA = Y
1	EA = X1 + Y
2	EA = X2 + Y
3	EA = X1 + X2 + Y
4	EA is obtained via indirect through Y
5	EA is obtained via indirect through Y + X1
6	EA = X3 + Y
7	Immediate Addressing

Bits 8 - 4 specify the instruction code. For instructions having the capability of being written in either short or long format, the bit configuration in Positions 8 - 4 of the long instruction is identical to that of the same opcodes in Bits 16 - 12 of its short format counterpart.

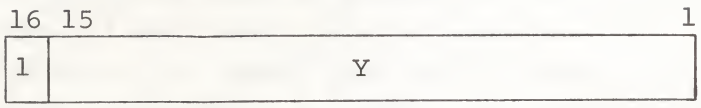
Bits 3 - 1 represent the designator bits which may be used for different purposes in different instructions.

The second word of a long format instruction contains:

1. A 16-bit two's complement value when immediate addressing is used. The sign is specified in Bit 16.

2. A 15-bit operand address. Bit 16 is ignored.

2.3 Indirect Addressing



The word to which a long or short format indirect instruction refers is of the above form. If Bit 16 of this word is zero, the effective operand address of the original instruction is contained in Bits 15 - 1. If Bit 16 is a "1", Bits 15 - 1 refer to another word of the above form wherein Bit 16 is again interpreted as the indirect bit.

2.4 Immediate Addressing

Immediate addressing refers to the use of the raw operand address as the operand of the instruction.

2.5 Skipping

Whenever an instruction leads to skipping of the next instruction the contents of register P will be incremented by one. This means that the skipped instruction must be in the short format. The skipped instruction will usually be a jump instruction.

2.6 Instructions

In the following list of instructions, S refers to Short Format and L refers to the Long Format.

Load/Store Operations

LDA: Load A (10 short or long)
Place the operand in Register A. The operand in memory remains unchanged.

LDE: Load E (11 short or long)
Place the operand in Register E. The operand in memory remains unchanged.

DLD: Double Load AE (14 short or long)

Place the operand in Register A and the contents of the memory location sequentially following the effective operand address in Register E. The contents of both locations in memory remain unchanged. The G bits in the Long Format may also specify set or clear memory protect for either of both memory locations if the Monitor Mode (MM) Flip-Flop is set. If MM is not set when a change in memory protection is specified, the memory remains unchanged and a memory protect violation interrupt will occur.

STA: Store A (22 short or long)

Replace the operand with the contents of Register A. Register A remains unchanged. Short format immediate addressing is forbidden. Long format immediate addressing will cause A to be stored in the second half of the instructions.

STE: Store E (23 short or long)

Replace the operand with the contents of Register E. Register E remains unchanged. Short format immediate addressing is forbidden. Long format immediate addressing causes E to be stored in the second instruction location.

DST: Double Store AE (20 short or long)

Store the contents of Register A in the memory location specified by the effective address. Store the contents of Register E in the next sequential memory location following the effective address. Immediate addressing is forbidden.

Arithmetic Operations

ADD: Add (15 short or long)

Add the contents of Register A to the operand and place the sum in Register A. The operand remains unchanged.

Add Overflow occurs if the sum exceeds 15 bits of magnitude. In this case, the sum is correct but the sign bit is not correct. An AO interrupt will occur whenever add overflow occurs if the program is not already in a higher priority interrupt routine.

SUB: Subtract (24 short or long)

Subtract the operand from the contents of Register A and place the difference in Register A. The operand remains unchanged.

Add Overflow occurs if the difference exceeds 15 bits of magnitude. In this case, the difference is correct, but the sign bit is not correct.

An AO interrupt will occur whenever add overflow occurs unless the program is already in a higher priority interrupt routine.

MPY: Multiply (16 short or long)

Multiply the contents of Register A by the operand and place the product in Registers A and E with the most significant bits of the product in Register A and the least significant bits in Register E. The product appears as a 30 bit signed number with A16 and E16 set to the correct sign. Octal (100000) X (100000) is an overflow condition which leaves both A and E equal to Octal (1100000). An AO interrupt will occur in this case unless the program is already in a higher priority interrupt routine.

DAD: Double Add (21 short and long)

Add the double-precision contents of Registers A and E and the double-precision contents of the memory location specified by the operand address and the next sequential memory location. The result appears as a 30-bit number in Registers A15 - A1 and E15 - E1 with the sign of the result in A16. The addend and augend should be two 30-bit numbers in the same format as the result with the signs in the upper halves of the numbers.

Add Overflow occurs if the sum exceeds 30 bits of magnitude. In this case, the sum is correct but the sign bit is not correct. An AO Interrupt will occur whenever add overflow occurs unless the program is already in a higher priority interrupt routine.

DVD: Divide (30 short or long)

Divide the 30-bit signed fraction contained in Registers A and E by the operand and place the quotient in Register E and the remainder in Register A. A true remainder is generated with the sign algebraically correct.

For proper division to occur, the dividend should have its most significant half in A15 - A1, and its least significant half in E15 - E1, with the correct sign appearing in A16. E16 is ignored.

Before division, if the absolute value of Register A is greater than or equal to the absolute value of the operand, a Divide Fault will occur. If Divide Fault occurs, the original contents of Registers A and E will remain unchanged, and the next instruction will be taken in sequence. If Divide Fault does not occur, the next instruction will be skipped, and the instruction following will be taken.

Logical Operations

LOR: Logical OR (12 short or long)
Form the logical OR of the contents of Register E and the operand and place the result in Register A. The original contents of Register E remain unchanged.

AND: Logical AND (13 short or long)
Form the logical AND of the contents of Register E and the operand and place the result in Register A. The original contents of Register E remain unchanged.

XOR: Exclusive OR (31 short or long)
Form the exclusive OR of the contents of Register E and the operand and place the result in Register A. The original contents of Register E remain unchanged.

Jump/Branch/Skip Operations

JMP: Jump (00 short or long)
Transfer control to the effective operand address. Short format immediate addressing is forbidden, Long format immediate addressing transfers control to the second instruction location.

JCI: Jump and Clear Interrupt (01 short or long)
Allow Interrupt.

Clear the highest-priority interrupt flip-flop that is set. If none is set, clear the nonpriority interrupt flip-flop. Take the next

instruction from the location specified by the effective operand address. Short format immediate addressing is forbidden. Long format immediate addressing transfers control to the second instruction location.

JLZ: Jump if A< 0 (07 short or long)
If the contents of Register A is algebraically less than zero, take the next instruction from the location specified by the effective operand address. Immediate addressing is forbidden.

JEZ: Jump if A = 0 (06 short or long)
If the contents of Register A equals zero, take the next instruction from the location specified by the effective operand address. Immediate addressing is forbidden.

JGZ: Jump if A >0 (05 short or long)
If the contents of Register A is algebraically greater than zero, take the next instruction from the location specified by the effective operand address. Immediate addressing is forbidden.

XEC: Execute (03 short or long)
Execute the instruction in the memory location that is specified by the effective operand address but do not transfer control unless the addressed instruction does so. If the specified instruction requires a skip, the location following the XEC Instruction will be skipped.

KAL: SKIP A LOW (25 short or long)
If the contents of Register A is algebraically less than the operand, skip the next instruction. If not, take the next instruction in sequence.

KAH: SKIP A HIGH (27 short or long)
If the contents of Register A is algebraically greater than the operand, skip one instruction. If not, take the next instruction in sequence.

KAQ: SKIP "A" EQUAL (26 short or long)

If the contents of Register A is equal to the operand, skip one instruction. If not, take the next instruction in sequence.

IMT: Increment Memory and Test (17 short or long)

Increment the operand by one (Short Format) or by the specified increment (Long Format) and skip if the operand becomes zero or if its sign changes G bits are used as the increment in Long Format. The contents of P + 1 will be incremented and tested if immediate addressing is specified in the long format.

RTJ: Store Return Address and Jump (02 short or long)

Place the address of the instruction sequentially following the RTJ Instruction in the operand. Take the next instruction from the location sequentially following the location specified by the effective operand address.

JTB: Jump and Test Bits (04 short or long)

Transfer control to the address that is contained in memory location (F + K). F is the effective operand address of this instruction. K is the number of left shifts that are required to shift the left-most one in Register A through A16. If Register A contains all zeros, K is zero. Register A is unchanged except that the left-most one is cleared. Register E will be changed to 100000.

X10: EXTERNAL INPUT/OUTPUT (36 short only)

The X bits specify what is to be done. D6 - D1 specify the external device (ED) address of the device to which the instruction is directed. Only the addressed device will respond. If D7 is a one, the operation specified by X occurs regardless of the busy status of the addressed device and no skip will occur. If D7 is a zero, the operation specified by X will occur and the next instruction will be skipped only if the addressed device is not busy. If D8 is a one, the specified operation is a control operation unless X = 2 (X = 2 is always a control operation). If D8 is a zero, the operation is a data transfer operation unless X = 2. The following operations may be specified in the X bits:

X	Operation
0	Send the contents of P + 1 out on the I/O lines and take the next instruction from P + 2 or P + 3 if a skip is required.
1	Send the contents of the location whose address is in P + 1 out on the I/O lines and take the next instruction from P + 2 or P + 3 if a skip is required.
2	Send zero on the I/O lines 16 through 2. Send D8 on I/O line 1. Take the next instruction from P + 1 or P + 2 if a skip is required.
3	Send the contents of Register A out on the I/O lines and take the next instruction from P + 1 or P + 2 if a skip is required.
4	The addressed device puts its information on the I/O lines. The result goes to P + 1. Take the next instruction from P + 2 or P + 3 if a skip is required.
5	The addressed device puts its information on the I/O lines. The result goes to the memory location whose address is in P + 1. Take the next instruction from P + 2 or P + 3 if a skip is required.
6	Send a STOP signal to the addressed device and take the next instruction from P + 1 or P + 2 if a skip is required.
7	The addressed device puts its information on the I/O lines. The result goes to Register A. Take the next instruction from P + 1 or P + 2 if a skip is required.

Index Operations

LDX: Load Index (10 or 11 long only with G = 1,2,3)
Place the operand of this instruction in the specified Index Register.

STX: Store Index (22 or 23 long only with G = 1,2,3)
Store the designated Index Register in the location specified by the operand address.

AXD: Augment Index by Displacement and Test (33 short only)
Augment the index register specified by X by the twos complement value of D. If the result is larger than the contents of P + 1, take the next instruction from P + 3, otherwise take the next instruction from P + 2.

Shift Operations K = D5 through D1 (34 short only; bits 11-7 designate the type of shift)

RSA: Right Shift A
Shift Bits 15 through 1 of Register A right k places. All bits shifted out of A, will be lost and A16 will be entered into A15. Sign Bit 16 of Register A is unchanged. The original contents of Register E remain unchanged.

RSD: Right Shift AE
Shift Bits 15 through 1 of Registers A and E right k places. All bits shifted out of A1 will be placed into E15 and all bits shifted out of E1 will be lost. A16 will be entered into A15. Sign Bits of Registers A and E are unchanged.

RLA: Logical Right Shift A
Shift the contents of Register A right k places. All bits shifted out of A1 will be lost and "0's" will be entered into A16. The original contents of Register E remain unchanged.

RLD: Logical Right Shift AE
Shift the contents of Registers A and E right k places. All bits shifted out of A1 will be placed in E16. All bits shifted out of E1 will be lost and "0's" will be entered into A16.

TRP: Transpose Shift

Shift the contents of register A right k places and the contents of register E left k places. A1 goes to E1 and E16 goes to A16.

LSA: Left Shift A

Shift Bits 15 through 1 of Register A left k places. All bits shifted out of A15 will be lost and zero will be entered into A1. Sign Bit of Register A is unchanged. The original contents or Register E remain unchanged.

LSD: Left Shift AE

Shift Bits 15 through 1 of Registers A and E left k places. All bits shifted out of E15 will be placed into A1 and all bits shifted out of A15 will be lost. Zero will be entered into E1. Sign Bits of Registers A and E are unchanged.

LLA: Logical Left Shift A

Shift the contents of Register A left k places. All bits shifted out of A16 will be lost and "0's" will be entered into A1. The original contents of Register E remain unchanged.

LLD: Logical Left Shift AE

Shift the contents of Registers A and E left k places. All bits shifted out of E16 will be placed into A1. All bits shifted out of A16 will be lost and "0's" will be entered into E1.

CSA: Circular Left Shift A

Shift the contents of Register A left circular k places. All bits shifted out of A16 will be placed into A1. Sign bit A16 is shifted with the contents of Register A. The original contents of Register E remain unchanged.

CSD: Circular Left Shift AE

Shift the contents of A and E left circular k places. All bits shifted out of A16 will be placed in E1 and all bits shifted out of E16 will be placed in A1. Sign bits of Registers A and E are shifted with the contents of the registers.

NORM: Normalize

Shift Bits 15 through 1 of Registers A and E left until A15 is equal to A16 or until 30 shifts have been performed. All bits shifted out of E15 will be placed in A1 and zero will be entered into E1. Sign bits of Registers A and E are unchanged. The shift count is placed in Index Register 1.

TAL: Tally

Rotate A left and count ones. The tally count is placed in Index Register 1. Register E will be changed to octal 100000.

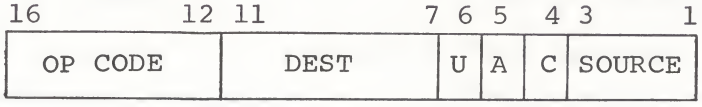
NOP: No Operation

Shift zero places.

Register Change

MOV: Move (35 short only)

The MOV format is as follows:



No address modification will occur.

OP CODE - Identifies the move instruction.

STEP 1 - Select the origination of the information that is to be moved.

0 - Zero
1 - X1
2 - X2
3 - X3
4,6 - A
5,7 - E

STEP 2

If C is a "1", the selected information will be complemented.

STEP 3

If A is a "1", add one to the result of the previous step. This step will not result in an add overflow.

STEP 4

Transfer the result of the previous step to the destination or destinations specified by the DEST bits.

BIT 11 specifies X3
BIT 10 specifies X2
BIT 9 specifies X1
BIT 8 specifies A
BIT 7 specifies E

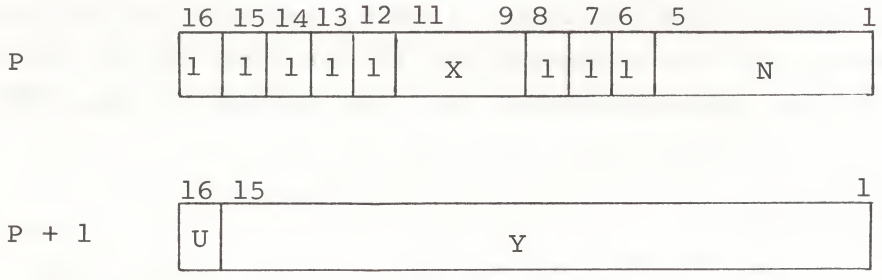
} Any combination may be specified.

U is Unused.

Programmed Instructions

PIN: Programmed Instructions (34,35,36, or 37 long only)

The format for PIN as follows:



Bits 16 through 12 are all ones (specifies Long Format).

X specifies address modification.

Bits 8 through 6 are all ones (identifies programmed instruction).

N is the number of the programmed instruction.

U is unused.

Y is the raw operand address.

Obtain the Subroutine Entry Address, SE, from Memory Location (00040 + N). Store the effective operand address in SE. Store (P + 2) in (SE + 1). Transfer control to SE + 2. This instruction calls sub-routines in a general way with a minimum of memory locations. It also provides an efficient means of passing the effective address of a single-length or multiple-length parameter to the subroutine along with a return address.

Memory Protect Operations

SMM: (00 or 01 long only, G=6,7)
Set Monitor Mode Flip-Flop and jump to 00000.

CMM: (00 or 01 long only, G=4,5)
Clear Monitor Mode Flip-Flop and jump to operand. The Monitor Mode Flip-Flop (MM) allows changing of memory protect status when it is set. SMM and CMM are suboperations of the Long Format JMP Instruction. The G bits specify whether the instruction is SMM, CMM OR JMP.

SMP: (14 long only, G=3,5,7)
Set Memory Protect.

CMP: (14 long only, G=2,4,6)
Clear Memory Protect. The memory protect operations are suboperations of the Long Format DLD command, i.e. the G bits of this Long Format command may specify SMP or CMP. Memory protect commands will operate on either or both of the same two operands a DLD command would. Thus, memory protection may be assigned to two locations with one instruction. The A and E Registers are loaded whether G specifies set/clear memory protect or not. The memory

protection will be changed only if the Monitor Mode Flip-Flop (MM) is set. If MM is not set when SMP or CMP occurs, a memory protect violation interrupt will result unless a higher priority interrupt routine is in progress.

Status Operations

SST — Store Status (33 short, X = 7, D1 = 0)

Let PUSH be the address contained in the push-down fixed memory location. Store P in PUSH, A in (PUSH+1), E in (PUSH+2), X1 in (PUSH+3), X2 in (PUSH+4), and X3 in (PUSH+5). Store (PUSH+6) in the push-down fixed memory location. Take the next instruction from the memory location following the location of this instruction (even if this instruction results directly from an XEC instruction).

LST — Load Status (33 short, X = 7, D1 = 1)

Let PUSH be the address contained in the push-down fixed memory location. Load X3 with the contents of (PUSH-1), X2 with the contents of (PUSH-2), X1 with the contents of (PUSH -3), E with the contents of (PUSH-4), A with the contents of (PUSH-5), and P with the contents of (PUSH-6). Store (PUSH-6) in the push-down fixed memory location. This will result in the restoration of control and status to what it was before the previous SST instruction.

Two fixed PUSH locations are in each 6130. 00276 responds only to protected SST and LST instructions. 00277 is used for all unprotected SST and LST instruction. A protect violation interrupt will result if a protected SST or LST addresses unprotected memory or if an unprotected SST or LST addresses protected memory. This permits the program to detect when the limits of the push-down store have been violated.

The push-down stores may also be used for dynamic storage in reentrant subroutines.

SECTION THREE PHYSICAL DESCRIPTION

The ADVANCE 6130 computer is housed in a single, standard 24 inch cabinet. As mentioned previously when additional memory modules are added, options exceed the space available in the primary cabinet, a second cabinet of equal size is attached as part of the system. This standard cabinet provides both front and rear access for ease of maintenance.

The control console can be mounted as an end-mount or in a remote location. A standard work surface module and a special work surface holding the Teletype typewriter complete the components of the basic ADVANCE 6130 system. The illustration on the following page shows a basic ADVANCE 6130 Computer System with a paper tape system included.

SPECIFICATIONS:

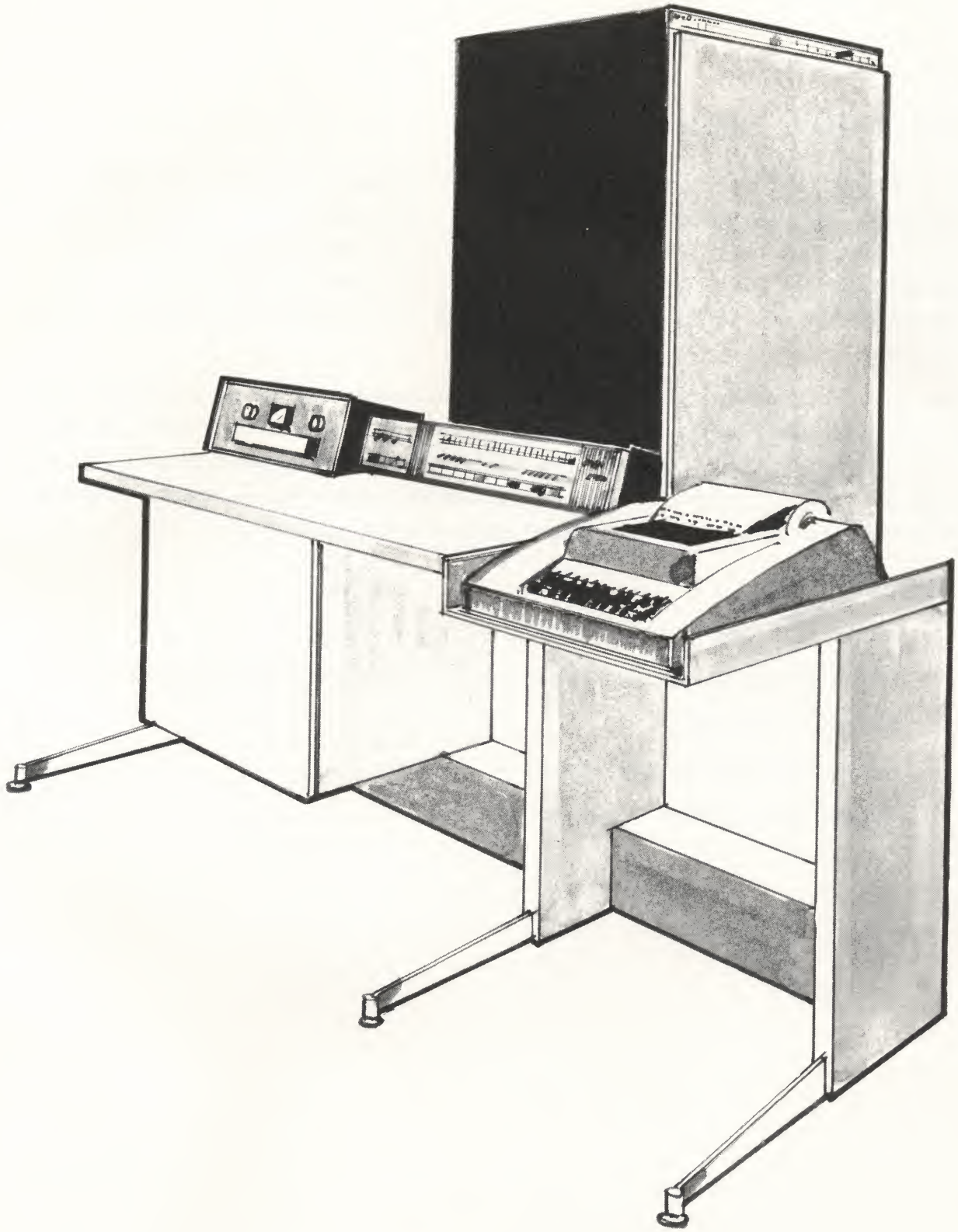
Basic Cabinet - 26"L x 26"W x 67.5"H

Additional Cabinet for memories and options (same as above)

Weight - 600 lbs.

Front and rear access required - 4 feet

Typewriter module - 24"L x 24"W x 29"H



Basic ADVANCE 6130 System Configuration

SECTION FOUR SOFTWARE SYSTEM

4.1 6130 Batch Process Operating System

A monitor, a loader and a set of Input/Output drivers will be available for the 4K computers.

An operating system will be available for the 8K and larger machines which will allow for batch processing, compile and execute, assembly and execute, linking and most of the ADVANCE Series operating system capabilities. The monitor is written as to allow for expansion, program recall, instantaneous control, and the ability to automatically overlay itself with real-time and other monitors.

4.2 6130 Assembler

Both a minimal assembler for 4K and an extended assembler for 8K and larger machines are available.

The 4K assembler will accept source input from cards, paper tape, or magnetic tape in symbolic form. Standard data allocation pseudo codes are available and constant declarations will be provided. A listing is available for optional use. Codes to provide for Programmed Instruction use and floating point macros will be provided.

The 4K assembler will operate with the minimum configuration which consists of a 4K central processor, a paper tape I/O, and a typewriter.

The extended assembler contains all the capabilities of the 4K assembler, plus programmer defined macros, operation code redefinition capability, common and link codes, listing control codes, IF statement control, address arithmetic, and other capabilities that currently exist in the ADVANCE Series assembler.

4.3 Compiler

A basic FORTRAN compiler is provided in the 6130 software package. The compiler performs its read/write functions through the use of Central I/O subroutine (CIO) which allows operator control of input/output peripherals and permits the compiler to be device independent. Features such as source listing and a diagnostic-only pass are provided on an optional basis. Extended features will include A-FORMAT and Triple Subscripting.

4.4 Library

Mathematical subroutines will operate on a 2 word floating point format or 3 word floating point format.

Most subroutines which involve parameter passage will be allocated as Programmed Instructions.

Provided with the basic compiler are certain non-standard library subroutines necessary for the execution of a Fortran object program, such as argument transfer routines, an alphanumeric formatter, an input/output routine, a binary input/ourput routine and other device manipulation subroutines.

4.5 Utilities

A basic set of utility routines for the 6130 includes alphanumeric and binary data transfer routines for card to tape etc., operations, routines for dumping and reloading core images on and from specified peripherals, a basic formatted binary loader and a snapshot core dump routine.

4.6 Real-Time Executive

Real-time capabilities will be available and will be defined at a later date.

ASI COMPUTER DIVISION

ASI COMPUTER DIVISION, ELECTRO-MECHANICAL RESEARCH, INC., 8001 BLOOMINGTON FREEWAY, MINNEAPOLIS, MINN. 55420

SALES OFFICES

BOSTON 617-272-5400, 1 Garfield Circle, Burlington
CHICAGO 312-323-0450, Suite 208, 119 E. Ogden Ave., Hinsdale, Ill.
DALLAS 214-357-4086, Suite 1329 Frito-Lay Towers
DENVER 303-789-1834, 3600 S. Lincoln, Englewood
EL PASO 915-751-2344, 8888 Dyer St.
HOUSTON 713-225-1811 1900 Southwest Tower
HUNTSVILLE, ALA. 205-881-4822, 3322 S. Memorial Parkway
LOS ANGELES 213-782-7030, 15551 Cabrito Road, Van Nuys
MINNEAPOLIS 612-888-9581, 8001 Bloomington Freeway
NEW YORK CITY 212-679-5954, 475 Fifth Avenue
SUNNYVALE, CALIF. 408-245-3694, 505 West Olive
WASHINGTON, D.C. 301-864-6340, 5012 College Avenue, College Park, Md.